PROGRAMMA DEL DISCO RAM PER PC

Nelle pagine seguenti verrà esaminato il contenuto del file sorgente per la creazione di un disco RAM, e verrà proposto un esempio significativo per realizzarlo praticamente. Se si desidera entrare maggiormente nei dettagli di funzionamento, leggendo questo capitolo e sfruttando le informazioni contenute nella libreria Objex si potranno trovare delle indicazioni molto interessanti.

Un metodo per semplificare la creazione di programmi è quello di avere a disposizione una libreria contenente le routine più utilizzate

La programmazione costiste fondamentalmente nella capacità di combinare esperienze precedenti vissute con problemi simili e nuove soluzioni adottate per risolvere le difficoltà attuali; il risultato di questa operazione dovrebbe sfociare in una nuova serie di routine compatibili con il programma in questione, che devono essere opportunamente inserite. Per semplificare il processo di ricerca delle funzioni e delle routine necessarie, si sfrutta una libreria di

Se il PC è dotato di un controller SCSI, prima di installare la scheda del disco RAM bisogna verificare che indirizzo di memoria occupa

funzioni già esistente; in questa sono disponibili tutti i procedimenti e le subroutine associate, che potranno essere aggiunte al nuovo programma diventandone parte integrante. Generalmente un programmatore impiega la maggior parte del tempo, e dedica il massimo sforzo, allo studio del progetto strutturale del programma, più che ad ogni altra operazione. Un metodo per semplificare e facilitare notevolmente la produzione di nuovi programmi è quello di crearsi una libreria contenente le routine e i procedimenti più utilizzati; in questo modo è possibile recuperare di volta in volta le funzioni necessarie, modificandole opportunamente per renderle adatte ai nuovi programmi. Quando si aggiornano dei programmi obsoleti, si può facilmente constatare che molte delle strutture e delle caratteristiche principali degli stessi sono simili tra di loro, poiché il modo in cui vengono sviluppate dipende essenzialmente dallo stile personale di ciascun programmatore. Le librerie fisiche sono diretta funzione delle librerie personali o delle strutture mentali dei programmatori, che in pratica rappresentano il metodo con il quale ognuno di essi sviluppa il lavoro. Con il passare del tempo i programmi tendono ad essere sempre più relazionati; infatti, le nuove versioni sono sempre meno diverse dalle vecchie da cui derivano. Quando si confrontano programmi *autonomi*, o diversi, che operano in un ambiente di lavoro comune, generalmente è necessario effettuare un gran numero di modifiche per renderli mutuamente compatibili.

La libreria Objex contiene una collezione di funzioni a basso livello

LIBRERIE OBJEX

L'idea principale, e lo scopo, per cui viene fornita la libreria Objex, è quella di dotare l'utente di una libreria di funzioni e routine standard che possano essere utilizzate in programmi diversi. Ciò consente di avere a disposizione un ampio sistema con un alto livello di compatibilità, che può essere modificato o ridefinito con operazioni molto semplici.

La libreria Objex contiene una collezione completa di funzioni a basso livello, alcune scritte in linguaggio C e altre in Assembler, ed è stata

zione /s. Su questo floppy si deve copiare anche il file SK.COM.
Si da per scontato che il contenuto di PKMIN sia già stato installato nell'hard disk; se questa operazione non è stata eseguita, bisogna inserire il floppy che lo contiene nell'unità A:, entrare nella sottodirectory PKMIN, e digitare il comando PKMIN C:. In questo modo i file vengono scompattati e installati sul disco C:. Dopo aver eseguito questa operazione bisogna entrare nella sottodirectory creata dal processo di installazione e scrivere un file BATCH (ESEMPIO.BAT), che potrebbe essere il seguente:

```
echo off
echo PROGRAMMA ESEMPIO.BAT
echo Questo è un esempio di file batch
rem
echo per emulare nell'unità B:
echo tutti gli archivi
echo presenti nell'unità A:
echo utilizzando 2 memorie
echo RAM 62256 come unità B:
echo Gli archivi di uscita
echo saranno pkdisk.pk1
echo e pkdisk.pk2
rem
pk a: c\<subdir>\pkdisk /d:b /n:2 /t:62256
rem
```

Dove A: è l'unità che contiene il floppy con gli archivi che devono essere emulati durante il processo di esecuzione del programma PK. C:<subdir> rappresentano l'unità e il nome della sottodirectory nella quale vengono memorizzati gli archivi del simulatore.
Pkdisk è il nome dato agli archivi generati, quando si esegue il file PK; questi file contengono i valori esadecimali che devono essere memorizzati nelle memorie RAM o EPROM. PK2 è relativo ai valori di IC1, mentre PK1 a quelli di IC2. Il parametro /d:b rappresenta il nome dell'unità che si vuole emulare, in questo caso B:. Infine, /n:2 rappresenta il numero delle memorie presenti fisicamente sulla scheda, in questo caso 2, e /t:62256 indica il tipo di memorie installate. In realtà non sono queste le memorie effettivamente installate sul circuito, ma le 431000 o equivalenti; in ogni caso, dei 128 KByte che ognuna di queste ha disponibili è possibile occuparne esattamente solo 32.
A questo punto si deve inserire il disco sul quale era stato memorizzato il file SK.COM nell'unità A: ed eseguire il programma ESEMPIO.BAT scritto in precedenza.
A partire da questo momento viene avviato il processo che genera i dati necessari per le rispettive memorie utilizzate come disco RAM.
Terminato il processo si deve verificare che i file PKDISK.PK1 e PKDISK.PK2 creati dal processo precedente siano presenti. Questi due archivi devono trovarsi nella stessa directory nella quale sono presenti i file PK.EXE ed ESEMPIO.BAT, poiché sono i file master binari dell'emulatore del disco.
Se per emulare l'unità disco si vogliono utilizzare memorie EPROM, PROM, ecc., è necessario convertire i programmi scritti in linguaggio C o assembler nel formato INTEL, il formato più diffuso tra i programmatori di EPROM.

Pkdisk è il nome degli archivi che devono essere memorizzati nelle memorie RAM o EPROM

Il circuito può essere installato nel PC indipendentemente dalla scheda madre di cui è dotato

Come esempio viene installato nel disco RAM il programma SideKick

Conversione nel formato esadecimale INTEL

Per convertire questi due file master nel formato INTEL si consiglia di creare un file BATCH che contenga tutti i comandi necessari, in modo da non dover digitare ripetutamente le stesse istruzioni, con la conseguente maggior probabilità di commettere degli errori. Il programma BATCH potrebbe essere simile al seguente:

```
echo Esempio per generare i
echo file in formato INTEL
rem
echo Conversione del primo file
bin2hex <pkdisk.pk1 >pkdisk1.hex
rem
echo Conversione del secondo file
bin2hex <pkdisk.pk2 >pkdisk2.hex
rem
```

Dove pkdisk.pk1 è il primo file che deve essere convertito, e pkdisk1.hex è il file risultante dalla conversione nel formato INTEL. La stessa cosa vale per pkdisk.pk2 e pkdisk2.hex della penultima linea di programma; questi sono i parametri richiesti dal filtro bin2hex. Il nome degli archivi può essere modificato a proprio piacimento, ma si deve sempre prestare molta attenzione alla sintassi, poiché i simboli "<" e ">" sono parte integrante del parametro, così come lo spazio presente tra l'estensione .pk2 e il simbolo ">". Questi due file creati da bin2hex sono quelli interpretati dal programmatore di memorie EPROM. Con il file pkdisk1.hex si scrive la memoria IC2 e con pkdisk2.hex la IC1. Questi due file si possono salvare su disco per inviarli in un laboratorio attrezzato per la scrittura di due memorie EPROM 27512, se il disco che si desidera emulare è di tipo PROM; se invece si sceglie di emulare un disco RAM, questi due file devono essere copiati nelle memorie stesse (in questo caso le RAM con capacità equivalente alle 27512 sono le 62256). Poiché queste due memorie occupano 128 KByte (da D0000hex a E0000hex) della memoria del calcolatore, prima di installare la scheda nello slot di espansione bisogna verificare che questi indirizzi risultino liberi; se così non fosse, sarà necessario impostare i ponticelli nelle posizioni opportune per selezionare gli indirizzi liberi. Per caricare i file in formato INTEL agli indirizzi occupati dalle memorie del circuito si utilizza il programma DEBUG del DOS. Supponendo che il file pkdisk1.hex creato in precedenza contenga, in formato INTEL, l'unità da emulare, è sufficiente caricarlo con il DEBUG nel modo seguente:

```
DEBUG
N PKDISK1.HEX
L D000:000
```

e il programma verrà direttamente convertito in formato binario e memorizzato all'indirizzo indicato, che corrisponde alla memoria IC2 del disco RAM. Si deve ripetere questo procedimento per il secondo file, indicando però l'indirizzo **D800:000** occupato da IC1, poiché è CS0/ che abilita questa memoria (si osservi la mappa degli indirizzi e lo schema). Ovviamente, se si vogliono utilizzare degli indirizzi diversi è sufficiente cambiare le posizioni dei ponticelli sul circuito.
Da questo momento, quando si avvia il calcolatore la scheda montata sullo stesso corrisponde all'unità B:, anche se la macchina è dotata delle unità fisiche A: e B:; infatti, la seconda unità fisica viene completamente ignorata. Avviando il calcolatore è possibile verificare la rapidità con la quale viene eseguito il programma SideKick dalla nuova unità B.
Dopo aver imparato il modo con cui emulare l'unità B:, l'utente può esercitarsi ad emulare altre unità con altri programmi, o addirittura emulare l'unità A: con installato il sistema operativo.

creata indipendentemente dal sistema operativo DOS, in modo da poter eseguire e creare programmi ad un livello inferiore allo stesso; comprende il BIOS, gli aiuti di Debug, dei programmi per il controllo delle periferiche, ed altri programmi a livello di sistema. I programmi che devono operare nell'area di lavoro del DOS possono utilizzare sia le librerie tradizionali del linguaggio C, che la libreria Objex.

Quest'ultima libreria contiene inoltre le definizioni legate a una raccolta di strutture standard, che possono essere utilizzate e ampliate a proprio piacimento per una applicazione specifica. Le definizioni di base sono costanti, in modo che le diverse applicazioni nelle quali vengono utilizzate possano presentare molti aspetti in comune. Ciò è stato realizzato in forma trasparente grazie alla definizione di una intestazione che consente di essere estesa da parte dell'utente. Quando si sviluppano applicazioni simili, si possono aggiungere nuove librerie che le supportino.

BUS INDIRIZZI — PONTICELLI DI SELEZIONE

	A19	A18	A17	A16	A15	A14	32K BA 00		64K BA 01		128K BA 10	
C000	1	1	0	0	0	0	CS1	DC 00	CS1	DC 00	CS1	DC 00
C7FF	1	1	0	0	0	1						
C800	1	1	0	0	1	0	CS0					
CFFF	1	1	0	0	1	1						
D000	1	1	0	1	0	0	CS1	DC 01	CS0			
D7FF	1	1	0	1	0	1						
D800	1	1	0	1	1	0	CS0					
DFFF	1	1	0	1	1	1						
E000	1	1	1	0	0	0	CS1	DC 10	CS1	DC 10	CS0	
E7FF	1	1	1	0	0	1						
E800	1	1	1	0	1	0	CS0					
EFFF	1	1	1	0	1	1						
F000	1	1	1	1	0	0	CS1	DC 11	CS0			
F7FF	1	1	1	1	0	1						
F800	1	1	1	1	1	0	CS0					
FFFF	1	1	1	1	1	1						

Non si vuole annoiare il lettore con questa illustrazione, ma ricordare solamente che un errore nella selezione dell'indirizzo causa il non funzionamento di tutto il dispositivo

Queste librerie ausiliarie contengono procedure comuni ad una particolare categoria di applicazioni, e riducono la necessità della duplicazione di alcune funzioni.

MODIFICA DEL SOFTWARE

Se si desidera modificare il software per la creazione del disco RAM, è necessario eseguire l'installazione utilizzando il file *pkmax.bat*, che carica i programmi sorgenti nelle directory appropriate. Poiché tutte le applicazioni Objex usano i file contenuti nella libreria fornita (libreria di funzioni, strutture, file incorporati, ecc.), questi devono necessariamente essere disponibili. Gli ampliamenti di *ojlib.lib* devono essere eseguiti solamente se questi hanno valore universale, e le aggiunte o le modifiche che interessano il solo Simulatore dovranno essere eseguite nella sottodirectory *PKMAX*. Anche se i file sorgente completi e gli strumen-

I file sorgente e gli strumenti per la loro manipolazione sono forniti con la libreria

Come si può notare le memorie sono inserite negli zoccoli SmartWatch, e questi a loro volta negli zoccoli del circuito

Normalmente il controller per i dischi rigidi occupa gli indirizzi compresi tra C000 e D000

per familiarizzare con la loro struttura e comprenderne appieno il funzionamento. Come al solito, è necessario operare con una particolare attenzione salvando frequentemente e proteggendo il lavoro fatto e i nuovi file che sono stati creati. Il file *Pk.h* viene utilizzato per definire le estensioni dell'intestazione standard di Objex. Le caratteristiche dei programmi per la creazione dei dischi virtuali vengono inserite in questo file, così come la declamazione delle variabili per modulo che facilitano l'uso delle routine di Objex.

Il programma *pk.exe* ha il compito di generare i file master per gli emulatori di disco PROM o RAM. Questo programma richiede l'inserimento di alcuni dati, che devono essere specificati sulla linea dei comandi. Il file pk.exe assieme al modulo *pkrom.ojx* creato in precedenza, contiene il programma e le sue routine di installazione, e genera i file immagine PROM del programma di controllo e i dati che ottiene dall'ingresso del disco.

Il programma di installazione *pkinstal.c* viene rilevato durante la fase di ricerca in ROM eseguita all'avvio, e svolge le seguenti funzioni: riserva uno spazio di memoria RAM nel sistema per il programma principale, copia il programma nella memoria RAM, inizializza il programma e lo collega alla funzione di chiamata al disco.

Il programma *pkdriver.c* ha il compito di intercettare la chiamata al disco del BIOS, ed esegue una operazione equivalente di lettura/scrittura su una RAM o una PROM per emulare una unità disco. Per determinare il metodo di emulazione, il programma utilizza i parametri del disco immagine emulato e le specifiche di configurazione.

In pratica, si può notare che Objex è un importante metodo per permettere l'esecuzione di una notevole quantità di sistemi e di programmi applicativi. Grazie a questo è possibile semplificare il proget-

ti per manipolarli sono contenuti nelle librerie aggiuntive, è necessario lasciare queste ultime intatte; al contrario, è possibile modificare i file in esse contenuti per adattarli alle diverse applicazioni, copiando gli stessi in sottodirectory esterne. Nella sottodirectory *PKMAX* l'utente può operare con la massima libertà, per eseguire le variazioni che considera più opportune. Gli elementi più soggetti a modifiche possono essere il modulo di controllo, i tipi di configurazione supportati, o parti minori legate alla presentazione, come ad esempio la visualizzazione dei messaggi sullo schermo. Poiché questa versione del Simulatore non comprende il sistema di memoria paginata PROM, o a matrice paginata PROM, può diventare un interessante strumento per imparare ad eseguire delle modifiche. Per consentire al modulo di controllo l'esecuzione di queste funzioni bisogna utilizzare il file *pkdriver.c*, opportunamente modificato, lasciando però il programma originale intatto.

Il file *pk.* serve come guida per la costruzione automatica di altri moduli addizionali, e può essere modificato in funzione delle proprie esigenze.

Prima di cercare di eseguire qualsiasi modifica, è consigliabile studiare attentamente i file sorgenti dei programmi per la creazione del disco virtuale,

Il programma pk.exe ha il compito di generare i file master

to del software, poiché fornisce una struttura comune in grado di soddisfare quasi tutte le esigenze.

```
:1000000055AA08E87200CB0000000E00F0080000BE
:1000100050726F6D4B69740000000000000000001A
:10002000496E7374616C6C000000000000000000F9
:1000300000000000000000000000000000000000C0
:10004000010000000C000200200000000000000081
:1000500000000000000000000000000000000000A0
:10006000000002020100027000D002FD030009003E
:100070000200000000000000558BEC81EC1C0057D2
:1000800056B800008946E68946F68946FEE8A2068B
:100090008B5EFE8B4742D1E825010089 46F28B5EDC
:1000A000FE8A47022AE4B109D3E08946FA8B5EF65C
:1000B000837F0C007503E93D058B76F68B5EF603B6
:1000C000770C8976F68B46FA3946F67703E9030012
:1000D000E923058B5EF6837F0A0F7403E914058B11
:1000E0005EF68B470CB10AD3E8B10AD3E0050004F1
:1000F0008946F0BB00008EC3BB13048B46F0B10AE7
:10010000D3E8262907BB00008EC3BB1304268B0748
:10011000B106D3E08946F88B5EF6FF770CB8000095
:1001200050FF76F8FF76F6E8900550E8680583C43E
:100130000AB81600508B46E605400050FF76F88B53
:1001400046FE05400050E8710550E8490583C40AA1
:10015000B81300508B46E605610050FF76F88B46D9
```

Ecco come appare un file in formato INTEL quando lo si visualizza sullo schermo

IL FILTRO BIN2HEX

Tra le molte utility fornite, la più interessante è senza dubbio il filtro Bin2Hex, utilizzato per convertire un file binario nel formato esadecimale INTEL, che rappresenta un formato molto noto e utilizzato dai programmatori di PROM. Poiché si tratta di un filtro, l'uscita può essere indirizzata direttamente a un file, ad una porta di uscita, o allo schermo. La sintassi delle tre modalità è la seguente:

- bin2hex <pkdisk.pk1 >nome.hex
- bin2hex <pkdisk.pk1 >COM1
- bin2hex <pkdisk.pk1 >CON

Sicuramente la modalità più utilizzata è la conversione in un file esadecimale INTEL, poiché è l'unico che possa essere in qualche modo letto e compreso dal programmatore.
Per rendere più facile l'interpretazione, il file convertito in esadecimale viene suddiviso in righe e colonne. La prima colonna di tutte le file inizia con i due punti (:) seguiti da una serie di dati che vengono di seguito descritti. La struttura delle varie righe è sempre la stessa, anche se i dati sono diversi, per cui è sufficiente esaminare la prima. I primi due byte immediatamente successivi ai due punti indicano il numero di byte relativi ai dati presenti sulla linea; i quattro successivi indicano l'indirizzo corrispondente occupato da questi dati; i due successivi indicano il tipo di scrittura (nel caso in esame saranno sempre a zero). I byte successivi sono i dati espressi in formato esadecimale, oggetto della conversione effettuata, e saranno pari al numero indicato dai primi due byte che seguono i due punti di inizio riga. L'ultimo byte è la check-sum (somma di controllo), che viene calcolata sommando tutti i byte, iniziando da quello immediatamente successivo ai due punti sino all'ultimo, escludendo naturalmente dalla somma il byte proprio della check-sum. Il risultato ottenuto deve essere sottratto al valore FFhex e successivamente sommata una unità. Questa operazione può apparire strana, ma è semplicemente la trasformazione della somma ottenuta nel suo valore negativo convertito in complemento due.
Per chiarire eventuali dubbi si ricorre ad un esempio: si supponga che la prima linea del file esadecimale sia la seguente:
:01001000757A
Per facilitarne la comprensione la si divide nei blocchi indicati in precedenza:
:01——0010——00——75——7A
dove:
- 01 è il numero di byte della riga,

Il formato INTEL è utilizzato da molti programmatori di PROM

L'intestazione del programma sorgente bin2hex indica che è scritto in linguaggio Assembler

```
Ver ASCII: D:\...\pkmin\bin2hex.asm     Col 0        4,283 Bytes
;
; Filename: bin2hex.asm
;
;
; Functional Description:
;
;  This program converts the contents of a file to Intel hexadecimal format.
Since many Prom
;  Programmers will accept an Intel Hex download format, .BIN files may be
converted.
;  Filter outputs can be directed as shown in the examples below.
;
;
; Arguments:
;  bin2hex <infile.bin >outfile.hex
;  bin2hex <infile.bin >COM1
;  bin2hex <infile.bin >CON
;
; Return:
;
; Revisions
; 1.01 Added a set-segment-record for converting bin files over 64K
```

Nell'esempio in esame viene emulata l'unità B del calcolatore

La scheda del disco RAM può essere installata in uno qualsiasi degli slot liberi

- 0010 è l'indirizzo relativo occupato dal dato,
- 00 è il tipo di scrittura, che normalmente vale sempre 00,
- 75 è l'unico dato della linea,
- 7A è la check-sum.

Per verificare la check-sum si devono eseguire le seguenti operazioni:

01 + 00 + 10 + 00 + 75 = 86hex

Il complemento a due di meno ottantasei è:

FF - 86 = 79

al quale si deve sommare una unità; da questo il risultato finale 7A, che coincide con il valore indicato dagli ultimi due bit. É interessante conoscere questi dettagli relativi al file generato dal filtro *Bin2Hex*, poiché è quello che viene trasferito al programmatore di EPROM.

AVVIAMENTO DEL CIRCUITO

Inizialmente si deve mettere da parte per un istante la scheda dell'emulatore, poiché prima di installarla definitivamente nel computer bisogna compilare il software necessario e decidere quale programma installare e quale unità emulare. Ad esempio, si supponga di voler installare il programma SideKick, versione 1.52, che molti utenti hanno utilizzato per la sua semplicità di gestione e per le sue prestazioni; anche se esistono versioni più avanzate, quella sopracitata rimane ancora una delle più apprezzate. Questo esempio serve per descrivere il modo in cui è possibile realizzare il processo di emulazione di una unità disco contenente un programma applicativo. Nel caso in esame si emula l'unità disco B: del computer. A tal fine, la prima cosa da fare è quella di formattare un floppy con il sistema operativo in uso, senza trasferire i file di sistema, vale a dire senza l'op-

Per non dover digitare più volte i comandi relativi alle diverse verifiche che si devono eseguire, si può creare un file BATCH simile a questo

zione /s. Su questo floppy si deve copiare anche il file SK.COM.

Si da per scontato che il contenuto di PKMIN sia già stato installato nell'hard disk; se questa operazione non è stata eseguita, bisogna inserire il floppy che lo contiene nell'unità A:, entrare nella sottodirectory PKMIN, e digitare il comando PKMIN C:. In questo modo i file vengono scompattati e installati sul disco C:. Dopo aver eseguito questa operazione bisogna entrare nella sottodirectory creata dal processo di installazione e scrivere un file BATCH (ESEMPIO.BAT), che potrebbe essere il seguente:

```
echo off
echo PROGRAMMA ESEMPIO.BAT
echo Questo è un esempio di file batch
rem
echo per emulare nell'unità B:
echo tutti gli archivi
echo presenti nell'unità A:
echo utilizzando 2 memorie
echo RAM 62256 come unità B:
echo Gli archivi di uscita
echo saranno pkdisk.pk1
echo e pkdisk.pk2
rem
pk a: c\<subdir>\pkdisk /d:b /n:2 /t:62256
rem
```

Dove A: è l'unità che contiene il floppy con gli archivi che devono essere emulati durante il processo di esecuzione del programma PK. C:<subdir> rappresentano l'unità e il nome della sottodirectory nella quale vengono memorizzati gli archivi del simulatore.

Pkdisk è il nome dato agli archivi generati, quando si esegue il file PK; questi file contengono i valori esadecimali che devono essere memorizzati nelle memorie RAM o EPROM. PK2 è relativo ai valori di IC1, mentre PK1 a quelli di IC2. Il parametro /d:b rappresenta il nome dell'unità che si vuole emulare, in questo caso B:. Infine, /n:2 rappresenta il numero delle memorie presenti fisicamente sulla scheda, in questo caso 2, e /t:62256 indica il tipo di memorie installate. In realtà non sono queste le memorie effettivamente installate sul circuito, ma le 431000 o equivalenti; in ogni caso, dei 128 KByte che ognuna di queste ha disponibili è possibile occuparne esattamente solo 32.

A questo punto si deve inserire il disco sul quale era stato memorizzato il file SK.COM nell'unità A: ed eseguire il programma ESEMPIO.BAT scritto in precedenza.

A partire da questo momento viene avviato il processo che genera i dati necessari per le rispettive memorie utilizzate come disco RAM.

Terminato il processo si deve verificare che i file PKDISK.PK1 e PKDISK.PK2 creati dal processo precedente siano presenti. Questi due archivi devono trovarsi nella stessa directory nella quale sono presenti i file PK.EXE ed ESEMPIO.BAT, poiché sono i file master binari dell'emulatore del disco.

Se per emulare l'unità disco si vogliono utilizzare memorie EPROM, PROM, ecc., è necessario convertire i programmi scritti in linguaggio C o assembler nel formato INTEL, il formato più diffuso tra i programmatori di EPROM.

Pkdisk è il nome degli archivi che devono essere memorizzati nelle memorie RAM o EPROM

Il circuito può essere installato nel PC indipendentemente dalla scheda madre di cui è dotato

Come esempio viene installato nel disco RAM il programma SideKick

CONVERSIONE NEL FORMATO ESADECIMALE INTEL

Per convertire questi due file master nel formato INTEL si consiglia di creare un file BATCH che contenga tutti i comandi necessari, in modo da non dover digitare ripetutamente le stesse istruzioni, con la conseguente maggior probabilità di commettere degli errori. Il programma BATCH potrebbe essere simile al seguente:

```
echo Esempio per generare i
echo file in formato INTEL
rem
echo Conversione del primo file
bin2hex <pkdisk.pk1 >pkdisk1.hex
rem
echo Conversione del secondo file
bin2hex <pkdisk.pk2 >pkdisk2.hex
rem
```

Dove pkdisk.pk1 è il primo file che deve essere convertito, e pkdisk1.hex è il file risultante dalla conversione nel formato INTEL. La stessa cosa vale per pkdisk.pk2 e pkdisk2.hex della penultima linea di programma; questi sono i parametri richiesti dal filtro bin2hex. Il nome degli archivi può essere modificato a proprio piacimento, ma si deve sempre prestare molta attenzione alla sintassi, poiché i simboli "<" e ">" sono parte integrante del parametro, così come lo spazio presente tra l'estensione .pk2 e il simbolo ">". Questi due file creati da bin2hex sono quelli interpretati dal programmatore di memorie EPROM. Con il file pkdisk1.hex si scrive la memoria IC2 e con pkdisk2.hex la IC1. Questi due file si possono salvare su disco per inviarli in un laboratorio attrezzato per la scrittura di due memorie EPROM 27512, se il disco che si desidera emulare è di tipo PROM; se invece si sceglie di emulare un disco RAM, questi due file devono essere copiati nelle memorie stesse (in questo caso le RAM con capacità equivalente alle 27512 sono le 62256). Poiché queste due memorie occupano 128 KByte (da D0000hex a E0000hex) della memoria del calcolatore, prima di installare la scheda nello slot di espansione bisogna verificare che questi indirizzi risultino liberi; se così non fosse, sarà necessario impostare i ponticelli nelle posizioni opportune per selezionare gli indirizzi liberi. Per caricare i file in formato INTEL agli indirizzi occupati dalle memorie del circuito si utilizza il programma DEBUG del DOS. Supponendo che il file pkdisk1.hex creato in precedenza contenga, in formato INTEL, l'unità da emulare, è sufficiente caricarlo con il DEBUG nel modo seguente:

```
DEBUG
N PKDISK1.HEX
L D000:000
```

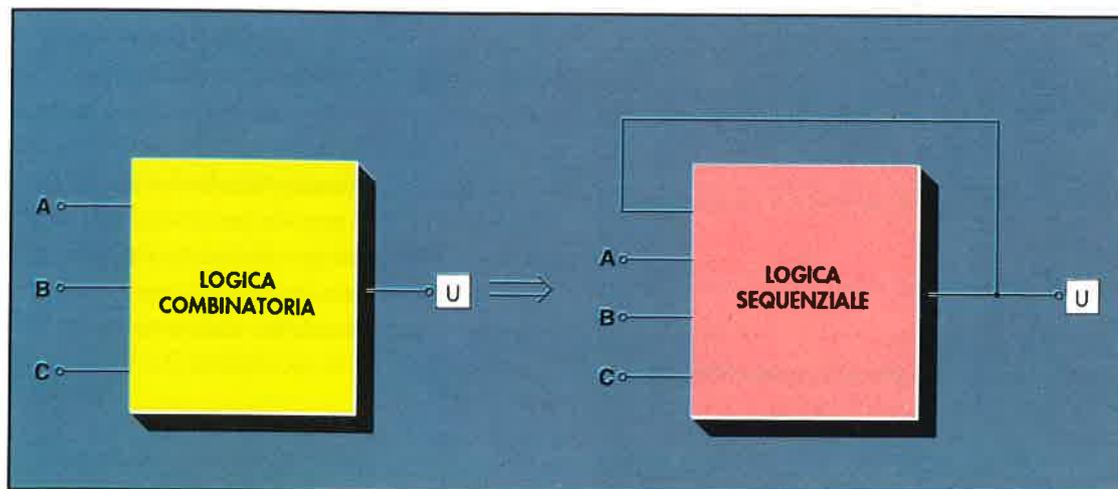
e il programma verrà direttamente convertito in formato binario e memorizzato all'indirizzo indicato, che corrisponde alla memoria IC2 del disco RAM. Si deve ripetere questo procedimento per il secondo file, indicando però l'indirizzo **D800:000** occupato da IC1, poiché è CS0/ che abilita questa memoria (si osservi la mappa degli indirizzi e lo schema). Ovviamente, se si vogliono utilizzare degli indirizzi diversi è sufficiente cambiare le posizioni dei ponticelli sul circuito.
Da questo momento, quando si avvia il calcolatore la scheda montata sullo stesso corrisponde all'unità B:, anche se la macchina è dotata delle unità fisiche A: e B:; infatti, la seconda unità fisica viene completamente ignorata. Avviando il calcolatore è possibile verificare la rapidità con la quale viene eseguito il programma SideKick dalla nuova unità B.
Dopo aver imparato il modo con cui emulare l'unità B:, l'utente può esercitarsi ad emulare altre unità con altri programmi, o addirittura emulare l'unità A: con installato il sistema operativo.

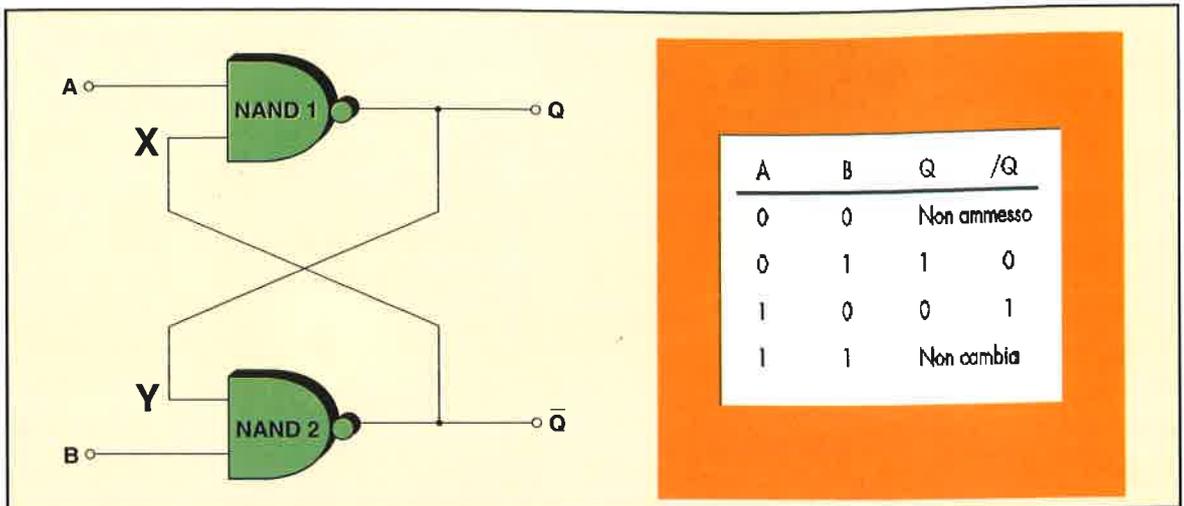
I CIRCUITI BISTABILI

Tutti i sistemi esaminati nei capitoli precedenti operavano con un tipo di logica definito combinatorio; i circuiti che vengono descritti di seguito funzionano invece con un tipo di logica detta sequenziale.

Un sistema sequenziale si diversifica da uno combinatorio per il fatto che le funzioni di uscita in ogni istante non sono solo funzione delle variabili di ingresso, come avviene per i circuiti combinatori, ma dipendono anche dallo stato in cui si trova il sistema.
Detto in altro modo, un circuito sequenziale si può definire come un sistema le cui variabili di ingresso dipendono da quelle proprie del circuito e dalle condizioni precedenti presenti in uscita.
Da questo si può dedurre che un sistema sequenziale reaziona le sequenze precedenti degli stati di ingresso in un modo determinato dalla sua struttura fisica, per cui può essere considerato un *sistema automatico*. La denominazione di *sequenziale* è dovuta al fatto che il valore della sua uscita, in un determinato istante, non dipende solamente dagli stati logici degli ingressi in quello stesso istante, ma anche dalla sequenza precedente degli stessi.

I CIRCUITI SEQUENZIALI

Per lo sviluppo dei circuiti sequenziali, e per poterne ricavare le equazioni di funzionamento come era stato fatto per la logica combinatoria, viene eseguito uno sdoppiamento delle variabili

La differenza tra logica sequenziale e logica combinatoria è legata alle variabili di ingresso che vengono prese in considerazione

A	B	Q	/Q
0	0	Non ammesso	
0	1	1	0
1	0	0	1
1	1	Non cambia	

Circuito sequenziale realizzato con porte NAND e tabella della verità corrispondente

di reazione introducendo un ritardo. Questo ritardo è realmente presente, in quanto rappresenta la somma dei tempi di ritardo di tutte le porte.
Considerando il circuito riportato nella figura corrispondente, formato da due porte NAND, le

Un sistema di tipo sequenziale si può progettare anche con porte NOR

uscite Q e /Q corrispondono a X e Y, ma con un ritardo intermedio. Tenendo presente questa condizione, per cui le funzioni di uscita vengono

Diagramma di una transizione tra stati stabili in funzione dell'impulso introdotto sull'ingresso B

ramificate in modo tale da poter diventare anche variabili di ingresso, è possibile ricavare le equazioni di sistema, che sono:

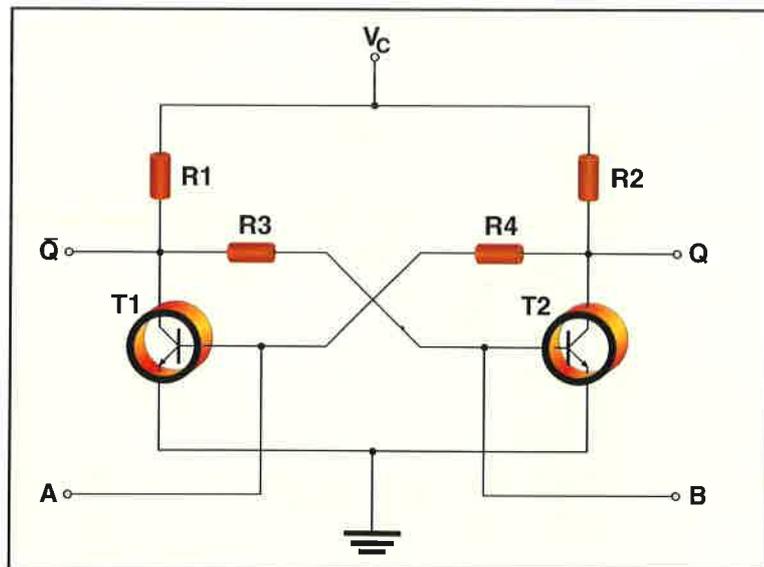
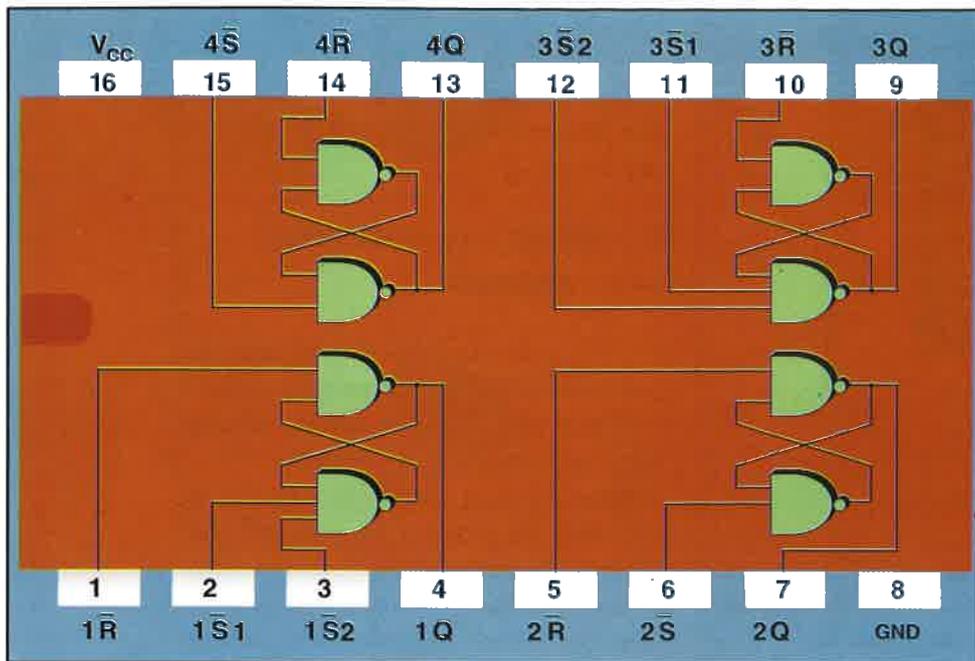
$Q = /(A \times X)$ e $/Q = /(B \times Y)$

In un circuito sequenziale i diversi stati delle uscite possono essere *stabili* o *instabili*. Infatti, il suo ciclo di lavoro è formato da una serie di fasi, di durata qualsiasi, durante le quali il sistema permane stabilmente finché una causa esterna (un ingresso) ne provoca la commutazione alla fase successiva. Queste fasi corrispondono alla condizione di stato stabile, mentre i periodi di transizione tra una fase e le successive rappresentano la condizione di stato instabile. Quest'ultima situazione si verifica in coincidenza di alcune particolari combinazioni dei valori di ingresso, mentre per altre le condizioni di uscita non sono accettabili. Ad esempio, la condizione per la quale entrambi gli ingressi si trovano al valore logico 0 non è accettabile, poichè provoca il passaggio sia di Q che di /Q al valore 1. Poiché le due uscite devono essere sempre complementari tra di loro, si instaurerà una situazione di indeterminazione che causerà la commutazione di una delle due in funzione delle condizioni interne precedenti e dei ritardi di propagazione dei segnali. Le condizioni ammesse sono quelle riportate nella relativa tabella della figura corrispondente. In conclusione, si può affermare che lo stato è stabile se alla configurazione raggiunta dalle uscite secondarie (X e Y) all'istante

La logica sequenziale sincrona è basata sul controllo della reazione tramite il clock

t, riportata all'ingresso all'istante t + t_D (con t_D che rappresenta il tempo di ritardo interno), corrrisponde la stessa configurazione delle uscite principali e secondarie per le stesse variabili di ingresso principali.
Nel circuito in esame l'equilibrio stabile si raggiunge quando nella tabella di funzionamento le variabili X e Y coincidono con le funzioni Q e /Q.
Quando si commuta una variabile di ingresso, il sistema passa da uno stato stabile ad un altro passando attraverso fasi adiacenti.
In un sistema reale non può commutare più di una variabile alla volta, per cui le transizioni tra fasi non adiacenti avvengono attraverso stati intermedi. Queste transizioni possono essere critiche o non critiche.
La transizione non risulta critica quando lo stato finale ottenuto è lo stesso, indipendentemente dal percorso fatto per raggiungerlo; è critica quando lo stato finale ottenuto è diverso.
I circuiti sequenziali analizzati sinora corrispondono ai circuiti asincroni.
Esistono anche circuiti sequenziali sincroni, nei quali la reazione delle uscite come variabili di ingresso è gestita da un clock, che consente la sua abilitazione solo a determinati intervalli regolari, come indicato nello schema corrispondente.
Il clock agisce quando gli ingressi relativi alla reazione corrispondono ai valori di uscita.

IL CIRCUITO FONDAMENTALE DI UN BISTABILE

Il multivibratore bistabile è un sistema che può assumere due condizioni di *equilibrio stabile* e una *instabile*. Questo circuito viene realizzato con dei transistori.
Inizialmente può sembrare che per la simmetria del circuito i due transistor conducano allo stesso modo: questa situazione corrisponde al punto di equilibrio instabile. Se per una qualsiasi ragione uno dei due transistor comincia a condurre, ad esempio T2, il valore della sua tensione di collettore diminuisce, abbassando di conseguenza anche la tensione di base di T1; ciò comporta un innalzamento della tensione di collettore di T1 che provoca un aumento della conduzione del transistor T2. Permanendo questa condizione si raggiunge l'istante in cui T2 entra in saturazione e T1 in interdizione, che corrisponde ad una delle due condizioni di equilibrio stabile. Ovviamente, l'altra condizione di equilibrio stabile si verifica con

Multivibratore bistabile convertito in un flip-flop grazie all'applicazione di impulsi esterni sulla base dei due transistor

Struttura interna di un circuito integrato convenzionale con quattro bistabili incorporati

T1 in saturazione e T2 in interdizione. Il multivibratore bistabile può oscillare da una posizione stabile all'altra grazie all'applicazione di impulsi esterni sulle basi dei transistor. Ritornando alla situazione precedentemente descritta, se si applica un impulso positivo sulla base di T1 quando questo transistor è in interdizione e T2 in saturazione, il primo inizia a condurre mandando in interdizione il secondo; ciò provoca il passaggio dalla prima alla seconda condizione di equilibrio stabile.

I bistabili sono generalmente conosciuti con il nome di *flip-flop*. Questi elementi vengono classificati come multivibratori bistabili, che possono assumere due condizioni di equilibrio stabile tra le quali vengono fatti oscillare tramite l'applicazione di impulsi esterni.

Il circuito fondamentale di un flip-flop realizzato con componenti analogici è rappresentato nella figura corrispondente, nella quale vengono indicati gli ingressi e le uscite. Se si indica con Q l'uscita di collettore di T2, il collettore del transistor T1 corrisponde di conseguenza all'uscita Q negata o complementata, indicata con /Q. Un impulso positivo applicato all'ingresso A, e perciò sulla base di T1, da origine ad un livello basso su /Q e ad un livello alto su Q, per cui l'ingresso A prende il nome di *Set* o *Preset*; nella maggior parte dei circuiti questo terminale viene rappresentato con la lettera *S*, e a volte viene definito anche come terminale di attivazione.

Un impulso positivo applicato all'ingresso B sulla base di T2 provoca come risultato la condizione di livello basso su Q e livello alto su /Q, da cui il nome di *Reset* o *Clear* dato a questo terminale. Nella documentazione tecnica e nei circuiti digitali con bistabili viene normalmente indicato con la lettera *R*, e a volte viene definito come terminale di azzeramento.

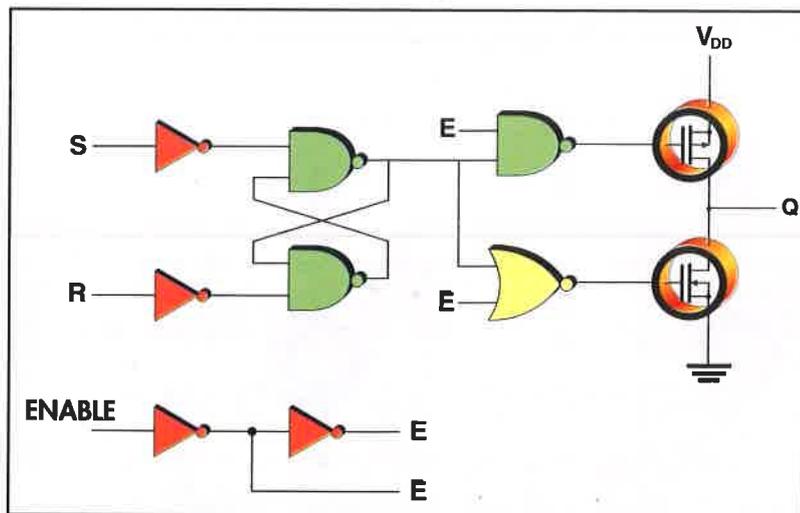
Nel capitolo successivo verranno esaminati i diversi tipi di bistabili che sono attualmente disponibili in commercio.

Elementi circuitali di un bistabile C-MOS

FOTO GENTILMENTE FORNITA DALLA DITTA MARCUCCI

TRASMISSIONE DI DATI TRAMITE LA RADIO

La necessità di trasmettere dati in qualsiasi momento e da qualsiasi luogo ha reso necessario l'impiego di sistemi e mezzi più complessi dei tradizionali metodi che prevedono l'utilizzo di un semplice cavo di connessione o di una linea telefonica.

Il controllo del flusso e la rapidità di spostamento dei dati (file di testi, disegni CAD, PCB, ecc.) sono diventati elementi molto importanti per chi opera con apparecchiature informatiche, considerando anche che la creazione di grandi banche dati ha generato un notevole aumento del traffico telematico tra elaboratori a queste collegate.
Anche i sistemi basati sui PC necessitano spesso di essere collegati tra di loro, ma per questi normalmente si ricorre a un semplice modem telefonico.

L'esistenza di grandi banche dati genera un notevole flusso di informazioni tra i calcolatori a queste collegati

FOTO MARCUCCI

Utilizzando i dispositivi per la trasmissione dei dati tramite radio si evita l'impiego delle linee telefoniche e si riducono i costi delle comunicazioni

Un sistema per la trasmissione dei dati via radio è costituito almeno da due trasmettitori-ricevitori

Tuttavia, non sempre è possibile allacciarsi ad una linea telefonica, sia perché esistono zone di difficile accesso, come ad esempio picchi montani, sia perché a volte è necessario collegare apparecchiature installate su mezzi in movimento, come ad esempio automobili e TIR.
In questi casi l'unica soluzione è quella di ricorrere alla trasmissione dei dati tramite radio, poiché solo in questo modo è possibile superare questo tipo di ostacoli.

VANTAGGI DELLA TRASMISSIONE VIA RADIO

La trasmissione dei dati via radio apre una nuova dimensione alle comunicazioni, e nel contempo presenta due grandi vantaggi rispetto ai metodi tradizionali:
- poiché è un'apparecchiatura completamente portatile, anche l'installazione risulta generalmente semplice e immediata, senza la necessità di realizzare collegamenti via cavo
- non dovendo pagare l'utilizzo di una linea è possibile rimanere collegati in modo permanente quando il tipo di lavoro lo richiede; l'unico costo è quello dell'energia che si consuma, che comunque è molto basso rispetto al costo di una linea telefonica dedicata.

STAZIONI MOBILI

Attualmente molti dei sistemi di trasmissione dati sono costituiti da stazioni mobili. Esempi tipici di questi sistemi sono rappresentati dalle apparecchiature utilizzate per la supervisione dei supermercati, dai dispositivi per la movimentazione dei carri ponte per il carico e lo scarico di navi commerciali e industriali, dai sistemi di messaggerie ottimizzati e da un'infinità di altre applicazioni. Si può facilmente notare che in tutti questi sistemi sarebbe molto costoso trasmettere i dati al computer centrale via cavo; utilizzando invece la comunicazione dei dati via radio, si ottiene una trasmissione in tempo reale che supera gli ostacoli legati alla mobilità delle apparecchiature, nella quale l'unico limite è costituito dalla portata del dispositivo. Con questo sistema di comunicazione è anche possibile realizzare una connessione multipla, formando una rete con diversi calcolatori collegati tra di loro.

Schema a blocchi di un trasmettitore di dati tramite radio, nel quale si possono osservare la parte radio, il modem e la sorgente di alimentazione

SISTEMA PER LA TRASMISSIONE DEI DATI VIA RADIO

Un sistema per la trasmissione dei dati via radio è costituito da almeno due trasmettitori-ricevitori; questo perché ciascuno dei due punti che si

desiderano collegare deve essere dotato almeno di un trasmettitore-ricevitore, composto da un trasmettitore, da un ricevitore, e da un MODEM (modulatore-demodulatore). Per rendere possibile questo processo è necessario avere a disposizione un programma di comunicazione adeguato al computer e al modem utilizzati.

Prima di iniziare ad operare bisogna però configurare i parametri di comunicazione tra il computer e il modem, che sono:

- *la velocità di trasmissione (1200, 2400, 4800, 9600, ..., baud)*
- *il tipo di parità*
- *il numero di bit della parola dato*
- *il numero dei bit di stop*
- *il protocollo di comunicazione*
- *il tipo di comunicazione (duplex o semiduplex).*

Configurando opportunamente il sistema è possibile inviare file di qualsiasi tipo, ricevere messaggi, eseguire programmi, ecc.

I dati elaborati dal PC o da un terminale (DTE) vengono trasmessi al modem attraverso una porta di comunicazione che generalmente corrisponde alla porta seriale. Questi dati vengono memorizzati nel buffer del modem, dove vengono opportunamente adattati: prima vengono impacchettati nel formato richiesto dal protocollo di comunicazione selezionato, e successivamente questi pacchetti vengono modulati in FSK (Frequency Shift Keying). Generalmente vengono utilizzati i canali audio del trasmettitore, che hanno una larghezza di banda simile a quella del canale telefonico che va da 300 a 3.400 Hz.

Dopo aver realizzato il collegamento via radio, e prima che il modem inizi a modulare in FSK, viene inviato al trasmettitore il segnale di inizio di trasmissione; in questo modo viene avviato il processo di trasmissione dei pacchetti di dati, che termina solo quando tutta l'informazione che si voleva inviare è stata trasmessa.

Il ricevitore cattura il segnale radio e ne effettua la demodulazione; il segnale demodulato viene inviato al modem, dove i dati vengono recuperati, registrati nel buffer, e trasmessi all'elaboratore attraverso la porta seriale RS232.

In una rete formata da diverse stazioni per la trasmissione-ricezione dei dati via radio è possibile che tutte siano in grado di ricevere le informazioni inviate da una stazione trasmittente; tramite un opportuno software di controllo però, i dati

I dati trasmessi dall'elaboratore o dal terminale (DTE) vengono inviati al modem tramite la porta seriale

Le uscite e gli ingressi del trasmettitore-ricevitore dei dati sono di tipo standard per facilitare la loro connessione ai differenti terminali (DTE)

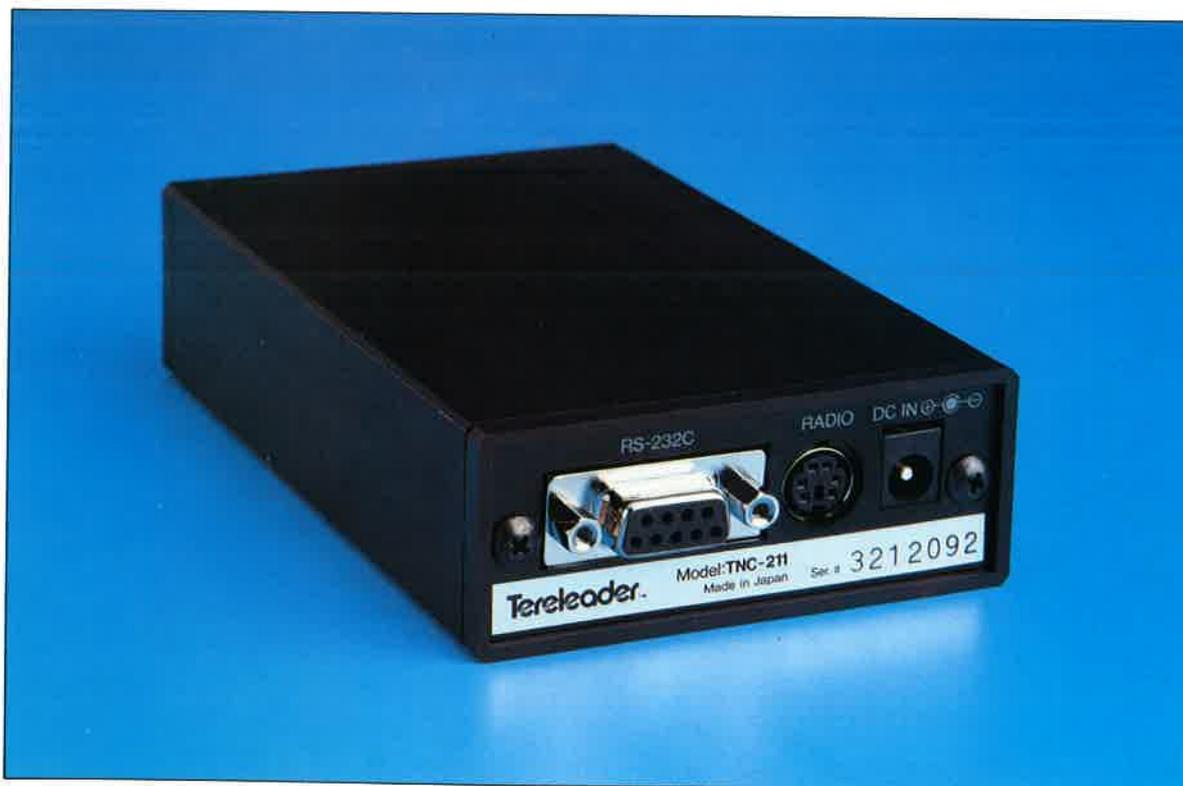



FOTO MARCUCCI

Per evitare errori di trasmissione, i modem sono dotati di una routine per la rilevazione degli stessi

possono essere inviati in modo selettivo solamente a determinate stazioni e non ad altre.

CORREZIONE DEGLI ERRORI

Nei sistemi per la trasmissione dei dati è molto importante fare in modo che le informazioni arrivino al destinatario correttamente e senza errori, poiché un difetto di trasmissione potrebbe causare grandissimi problemi e confusioni. Per evitare questi errori, i modem sono dotati di una routine di controllo della comunicazione che esegue una verifica dei dati dopo che questi sono giunti a destinazione; se viene rilevata una anomalia in ricezione, il ricevitore invia un comando al trasmettitore che gli impone di ritrasmettere il pacchetto dei dati finché questi non arrivano corretti. In questo modo viene garantita la qualità della comunicazione e l'affidabilità dei dati che arrivano al destinatario.

ANTENNA
AMPLIFICATORE DI POTENZA
MODULATORE
SEGNALE MODULANTE
MODEM
FREQUENZA PORTANTE
DATI

Il trasmettitore è costituito da un generatore di portante, da un modulatore che utilizza l'uscita del modem come segnale modulante, e dall'amplificatore finale di potenza

LA STAZIONE RADIO

Il sistema per la trasmissione-ricezione dei dati è composto dai seguenti elementi principali:

- *trasmettitore*
- *ricevitore*
- *modem*
- *antenna*
- *sorgente di alimentazione.*

- *TRASMETTITORE*: è il dispositivo incaricato di irradiare nello spazio, tramite l'antenna, il segnale ricavato dai dati forniti dal modem, ed è formato dalle seguenti parti:

* *modulatore:* riceve il segnale dal modem (modulato in FSK) e lo modula sulla frequenza portante della radio con il sistema di modulazione utilizzato da quest'ultima. I sistemi radio possono operare in modulazione di frequenza (FM), di ampiezza (AM) o di fase (PM), e nelle loro diverse varianti

* *generatore di portante:* fornisce il segnale portante che determina la frequenza di trasmissione

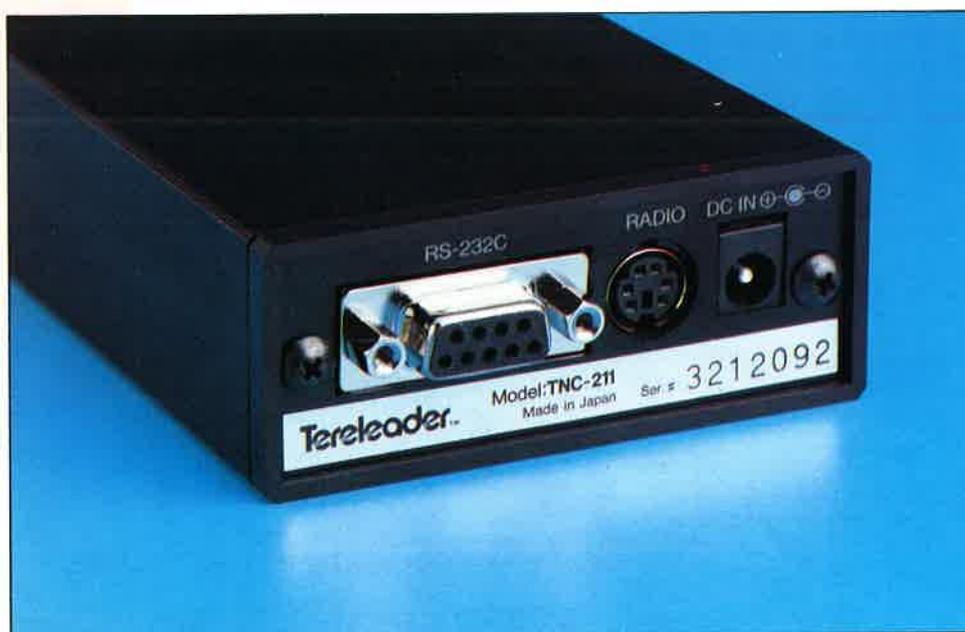
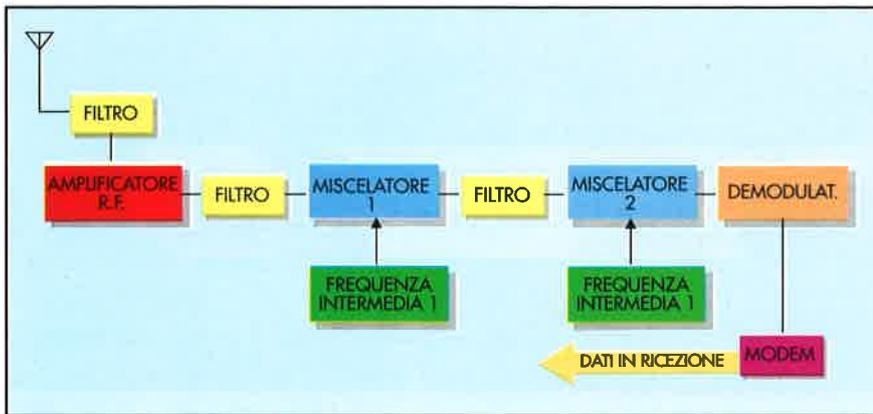
* *amplificatore di potenza:* fornisce il livello di potenza opportuno al segnale in radiofrequenza per poter essere trasmesso dall'antenna; il suo valore è funzione della portata richiesta

- *RICEVITORE*: capta il segnale irradiato nello spazio da un trasmettitore, lo demodula, e lo invia al modem; è costituito dalle seguenti parti:

* *stadio di ingresso:* la sensibilità di questo stadio è molto importante quando si deve definire la portata del collegamento, poiché amplifica il segnale e lo invia al miscelatore

* *miscelatore:* in questo stadio

La porta seriale RS232 viene utilizzata per collegare il trasmettitore dei dati al PC

FOTO MARCUCCI

Schema a blocchi di un ricevitore tradizionale, nel quale vengono impiegate due frequenze intermedie

vengono combinati il segnale ricevuto e quello proveniente dall'oscillatore locale, ottenendo in questo modo il segnale a frequenza intermedia che viene inviato al demodulatore

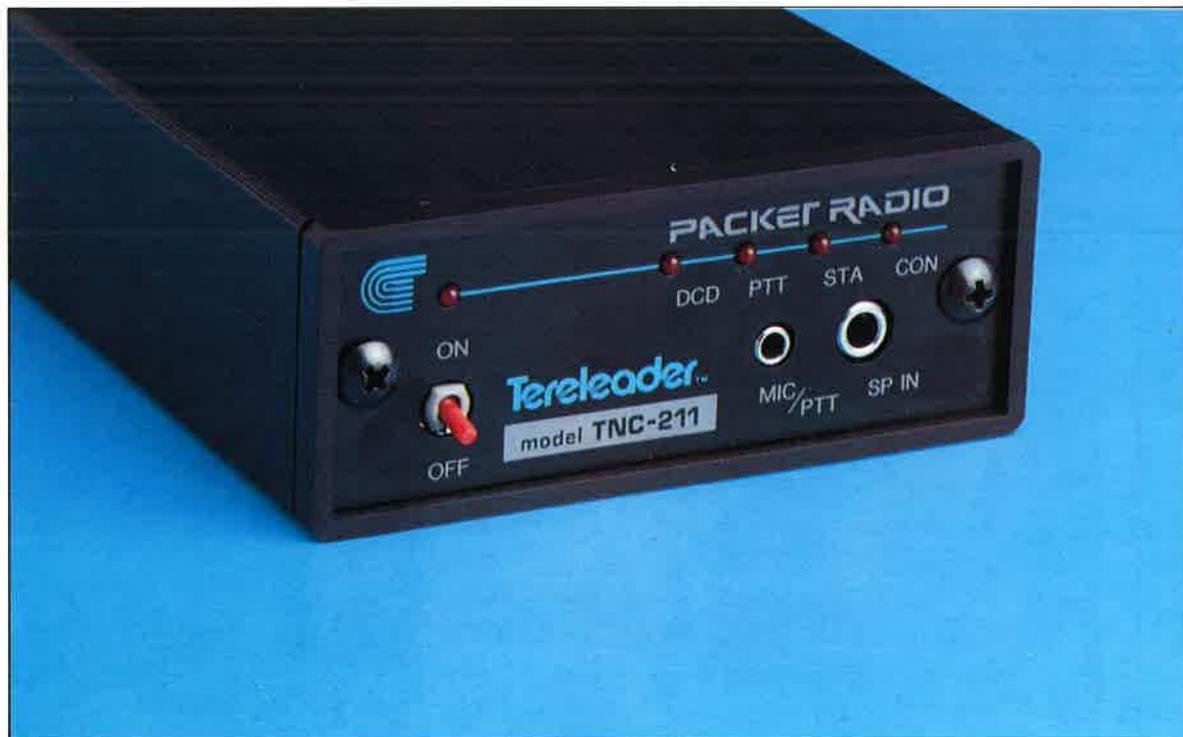
* *demodulatore:* in questo stadio, dal segnale a frequenza intermedia viene ricavato un segnale simile a quello che il modem del corrispondente invia al trasmettitore. Il segnale così ottenuto viene trasferito al modem, che ne estrae i dati per trasmetterli al computer.

- *MODEM*: è il dispositivo fisico che converte il segnale digitale ricevuto da un computer in un segnale analogico idoneo per essere trasmesso su di un canale audio; ovviamente deve essere in grado di svolgere anche la funzione inversa, vale a dire recuperare il segnale digitale partendo da quello analogico che arriva alla stazione ricevente, e trasmetterlo ad un PC o ad un sistema informatico.

Il modem serve da interfaccia tra il trasmettitore-ricevitore e il computer

La funzione del modem è quella di operare da interfaccia tra il trasmettitore-ricevitore e il computer: prima invia i segnali dei dati che arrivano dal computer al modulatore del trasmettitore, e successivamente recupera il segnale proveniente dal demodulatore del ricevitore per inviarlo al PC.
Il modem può essere controllato dall'elaboratore per mezzo dello stesso programma di comunicazione utilizzato per la trasmissione dei dati tramite la linea telefonica. Il modem deve comunicare sia con il computer, attraverso una porta seriale RS232, sia con la radio (trasmettitore-ricevitore), per cui ha bisogno di operare con almeno tre segnali: il segnale PTT (di attivazione del trasmettitore - quando non è attivo il dispositivo radio

Sul pannello frontale sono presenti delle segnalazioni che indicano lo stato di funzionamento dell'apparecchiatura

FOTO MARCUCCI

Il modem può essere gestito dall'elaboratore con lo stesso programma di comunicazione utilizzato per inviare i dati

Struttura convenzionale di una catena di dati, nella quale si possono osservare i vari campi che la compongono

rimane in ricezione), il segnale di ingresso al modulatore, e il segnale di uscita del ricevitore.

L'ALIMENTAZIONE

L'alimentazione a queste apparecchiature può essere fornita in diversi modi, che dipendono dal posto in cui sono installate.
I trasmettitori possono essere alimentati sia in corrente continua che in alternata, e l'impiego dell'una o dell'altra è funzione di ciò che si ha a disposizione nel luogo in cui si trova l'apparecchiatura. Infatti, se il dispositivo si trova in un ambiente chiuso dove è disponibile una presa collegata alla rete elettrica si può sfruttare questo tipo di alimentazione; in generale però, molti di questi dispositivi vengono installati su automobili o TIR, dove l'unica alimentazione disponibile è costituita dalla batteria per cui, grazie anche al loro basso assorbimento, vengono collegati a questa sorgente.

IL PERCORSO DEL SEGNALE

Il segnale seriale entra nel modem e viene caricato nel buffer di RAM, la cui capacità di memoria è generalmente di alcune decine di Kbyte; questo è a sua volta suddiviso in altri due buffer, uno di ingresso e uno di uscita, e in un blocco dedicato alla gestione del programma. La memoria RAM viene normalmente tamponata con una batteria, generalmente al litio, che permette il mantenimento delle informazioni nel caso venisse a mancare l'alimentazione. Dal buffer vengono prelevati i

Architettura tipica di comunicazione e controllo tra il modem e il PC